https://doi.org/10.31449/inf.v49i19.9935 Informatica **49** (2025) 425–438 **425**

Adaptive Hyperparameter Optimization for Financial Time Series Forecasting: A Chameleon Swarm-driven XGBoost Approach

Xia Xiao[1], Fang Wang[2*], Hongmei Xu[1], Dandan Wang[3], Yefeng Zhang[1]
[1]School of Economics and Management, Yantai Nanshan University, Yantai, Shandong, 265713, China
[2]Academic Affairs, Yantai Nanshan University, Yantai, Shandong, 265713, China
[3]Institute of higher education, Yantai Nanshan University, Yantai, Shandong, 265713, China
E-mail: szjjtdxx@163.com, 13176314345@163.com, daia-163@163.com, wangdandan129@163.com, 653842709@163.com
[*]Corresponding author

Keywords: financial time series, stock market prediction, financial data, weighted chameleon swarm-driven eXtreme gradient boosting (WCS-XGBoost), time series forecasting

Received: July 1, 2025

Time series forecasting is a central theme in the financial market, and the ability to estimate stock prices and trends accurately has a direct impact on investment strategies and risk management decisions. Statistical methods and neural network-based models often struggle to cope with the nonlinear and erratic nature of financial data. This work is aware of these shortcomings and proposes a new model, Weighted Chameleon Swarm-driven eXtreme Gradient Boosting (WCS-XGBoost), to improve prediction performance in challenging time series cases. Historical stock price data from credible public sources is collected, emphasizing daily closing prices and corresponding technical indicators. The data is normalized, then undergoes feature extraction via Principal Component Analysis (PCA) to reduce dimensionality while maintaining signal integrity. The predictive engine's central component, WCS-XGBoost, utilizes Chameleon Swarm Optimization to fine-tune XGBoost hyperparameters adaptively, maximizing accuracy and generalization. The model was implemented in python that exhibited superior performance with a Root Mean Square Error (RMSE) of 0.2312, accuracy of 98.69%, and Mean Absolute Percentage Error (MAPE) of 0.321, all of which were significantly better than baseline models. This framework highlights the potential of hybrid evolutionary learning in the advancement of stock market forecasting methodologies.

Povzetek: Članek obravnava izziv napovedovanja finančnih časovnih vrst, kjer klasični modeli slabo obvladujejo nelinearnost in volatilnost podatkov. Predlaga metodo WCS-XGBoost, ki združuje PCA za izbiro značilk, XGBoost za napovedni model ter bio-navdihnjeno optimizacijo Chameleon Swarm za prilagodljivo iskanje hiperparametrov. Model doseže bistveno nižji RMSE in visoko točnost napovedi.

1 Introduction

Financial markets' variables are complex, non-linear systems influenced by a wide selection of structures, including macroeconomic indicators, geopolitical developments, and institutional behaviour [1]. Time series models are well-suited for capturing these temporal dependencies and evolving dynamics, making them valuable tools for forecasting market trends such as price, returns, volume, and volatility [2-3]. These estimates enable investors, institutions, and regulators to make educated decisions and manage risk more efficiently. Time series prediction is important in financial market analysis, especially for projecting stock values based on previous trends. This process involves analyzing temporal patterns in past data to anticipate future movements [4]. Two main approaches guide stock market prediction: technical analysis, which focuses on identifying patterns in past prices, and fundamental analysis, which considers economic indicators, company performance, and management quality [5]. However, challenges remain due to the unpredictable and volatile nature, which often exhibits co-movement, seasonality, anomalies, and abrupt changes. The foundational assumption of technical analysis, in which patterns tend to repeat over time, underlines the relevance of time series models in modern financial prediction [6-7]. Traditional statistical and neural network-based models are often unsuccessful when predicting financial time series data because such data is nonlinear, volatile, and high-dimensional. Poor hyperparameter tuning and generalization limitations to authority are further issues when considering the forecasting performances to apply these techniques in real-world financial contexts. The objective is to enhance an effective time series prediction model for financial market data to improve stock data forecasting accuracy and support strategic decision-making under uncertainty.

1.1 Key contribution

- Research developed a stable and scalable financial time series forecasting model via hybrid AI methods to facilitate better market trend prediction and investment decision making.
- It uses a comprehensive stock market simulation dataset with real-world financial features that are generated using Geometric Brownian Motion and Markov Chains, which we have made available from Kaggle in the public domain. It standardizes the data using Z-score normalization and then employs Principal Component Analysis (PCA) to extract features and reduce dimensionality whilst retaining the necessary information contained within the signals.
- The proposed model further proposes a new variant called a Weighted Chameleon Swarm Optimization (WCS) with eXtreme Gradient Boosting (XGBoost), denoted in this research as the WCS-XGBoost algorithm. This variant optimized the hyperparameters, leading to improved predictive accuracy.
- Importantly, the WCS-XGBoost model captured the nonlinear designs and dependencies of financial time series price, which produced accurate forecasts of the stock trend.

1.2 Research objective and questions

Research is to create a better time series prediction model that focuses on improving accuracy in forecasting financial markets, especially stock price trends, through the implementation of the Chameleon Swarm Optimization approach and XGBoost into the WCS-XGBoost framework.

Research will seek to respond to the subsequent research questions:

- RQ1: How to create a hybrid evolutionary learning method to advance the accuracy of financial time series prediction?
- RQ2: How does the WCS-XGBoost model compare to the rigor and validity of traditional and Deep Learning (DL) models (i.e., LR, BiLSTM) in terms of accuracy, RMSE, MAPE, etc?
- RQ3: Can bio-inspired optimization (WCS) effectively tune the hyperparameters of the proposed models to create an adequate level of financial forecasting under conditions, in addition to volatility and non-linearity of the data?

The introduction was defined in Section 1, and the related work was detailed in Section 2; the methodological flow was identified in Section 3, the outcomes and discussion were reviewed in Section 4, and the research was concluded in Section 5.

2 Related work

Research developed a financial analysis and decision support system that complements enterprise financial decision-making in an uncertain environment using big data technologies [8]. It leveraged Hadoop and Spark to enable scalable processing of data, to analyze financial metrics, and also applied Net Present Value (NPV) and Internal Rate of Return (IRR) to evaluate different investments. The research development achieved an overall accuracy of 98.9%, and ratings for user satisfaction improved from 8.2 to 8.7. Here were four problems with the research: it did not have integration with real-time data sources, there was limited applicability by context through industry, there was reduced explainability, and applied validation efforts across enterprise scales and contexts were minimal. Research described a novel RedRVFL [9] system for forecasting financial time series. The methods used recurrent hidden layers had random initialization and fixed weights, which ensured stability throughout training. It used weighted hidden layers for DL, which extracted complicated patterns from datasets. Comparative results demonstrated greater forecasting accuracy and predictive capabilities. As a result of deep stacked layers and no weight tuning, the model overfitted to noise in a certain volatile financial dataset. The clustering approach utilized autoencoders to improve predictive model performance in the financial industry in the research [10]. The technique, which was applied to prominent financial data, demonstrated that using auto encoders improved the granularity and quality of clustering, successfully separating various groups of financial time series. More accurate financial forecasting models are made possible by changes to clustering algorithms, which offer helpful information for risk management and investing strategies. Research highlighted the possibility of advancing the accuracy of financial time series prediction with a hybrid method that incorporates CNN and polynomial regression. Chaos was first modeled in time series, then predictions were made using CNN, and polynomial regression was used to make output predictions using the residual errors [11]. The hybrid achieved a lower RMSE of 0.0021, MAPE of 1.92%, and Theil's U of 0.041, which outperformed Auto Regressive Integrated Moving Average (ARIMA), Random Forest (RF), Classification and Regression Tree (CART), and Chaos + CNN. The hybrid model required a lot of computational power, could not be generalized to data that was not chaotic, and the chaos detection and chaos modelling required some extensive parameter tuning. Table 1 shows the summary of recent research on financial marketing.

Table 1: Comprehensive analysis of existing models and techniques in financial prediction

Reference	Objective	Methodology	Findings	Limitations
Li et al., [12]	increased stock price forecasting accuracy with a new hybrid model.	K-means Clustering group sub-series with similar sample entropy for hierarchical decomposition and prediction.	Outperformed existing models across multiple indices; each component proved effective.	Complexity and computational cost hinder real-time applications.
Lazcano et al.,[13]	Improved oil price forecasting accuracy by combining GCN and BiLSTM models.	Developed a hybrid BiLSTM-GCN model for time series prediction.	Achieved lower RMSE, MSE, MAPE, and higher R^2 than individual models and traditional approaches.	Potential computational complexity and need for further validation on diverse datasets.
Albahli et al.,[14]	Stock market moves are forecasted using Google Finance and Twitter data.	Developed an extended opinion lexicon trained with ELM	Achieved 86.06% accuracy, outperforming existing methods.	Limited to Twitter and Google Finance data
Bu, [15]	Improved the precision and speed of financial decision making and risk forecasting using fuzzy logic, neural networks, and genetic algorithm approaches.	Used fuzzy logic to deal with uncertainty, Multilayer Perceptron (MLP) for nonlinear learning, and genetic algorithms to optimize financial decision support variables.	Budget overruns were decreased to 5%, ROI deviation to ±5%, risk warnings increased by 75–80%, and speed decreased to 30 seconds.	Had problems validating across different financial sectors, real-time dexterity, interpretability in black-box models, and large institutional scale.
Jiao, [16]	Attended to dynamic financial distress prediction by addressing concept drift problems with integrated Least Absolute Shrinkage and Selection Operator (LASSO) and Gradient Boosting Decision Tree algorithms (GBDT).	LASSO was utilized for feature selection, and GBDT for prediction, as well as a similarity index based on sample similarities, along with a time-series stratified sampling plan.	The model obtained 92.47% measured accuracy, 85.33% F-value, and 91.78% G-value, providing improvements across other classifiers and exhibiting consistency with resulting significance statistics.	There was a lack of generalizability to non-chain markets, reduced interpretability, a lack of sufficient multi-source data integration, and increased computational requirements for dynamic updates.
González-Núñez et al.,[17]	Developed an Artificial Organic Network for stock market prediction using the Index Tracking Problem.	Forecasted eight indices and compared performance with other ML methods.	Accomplished 0.9806 R-square, 7×10^{-4} error; showed ability to adapt, dynamic, and reconfigure topology across eight stock market indices.	Needs validation across diverse economic scenarios.

2.1 Research gap

In summary, there has been significant progress to date, but many current models for financial time series forecasting still have limitations related to real-time use, generalizability, and interpretability. For example, big data-driven systems are capable of highly accurate predictions (98.9%) using Hadoop and Spark [8], but are not capable of real-time integration or validation across sectors. DL applications, including RedRVFL [9] and BiLSTM-GCN [13], do provide high levels of prediction capability, but they are prone to overfitting, typically require significant computation resources, and are often not easy to explain. Autoencoder-based clustering methods also offer improvements in segmentation rate, but require significant tuning to use effectively [10]. Hybrid approaches have the potential to make significant improvements over ARIMA, RF, and CART, as shown using a chaos theory and CNN framework yielding less than 0.0021 RMSE [12], but are not only computationally expensive but also inflexible in application. Other approaches, such as those using fuzzy logic by [15], GBDT-LASSO by [16], and artificial organic networks by González-Núñez et al. (2024), show potential as well, but still lack types or domain robustness or have limited expansion utility. The problems highlighted in the content above represent an ongoing need for a scalable, nonstationary, and non-linear model which are effectively used and communicated across multiple sectors. This research proposed a model called WCS-XGBoost for financial forecasting, an interpretative framework that connects PCA-based dimensionality reduction with swarm-based hyperparameter optimization as a method for robust financial forecasting.

3 Methodology

Research presents a new hybrid Artificial Intelligence (AI) method to improve accuracy on time series forecasting in finance. The stock market data, utilizing daily closing prices and several important technical indicators is utilized. Data pre-processing, including feature scaling through normalization and reducing dimensionality through PCA, was confirmed to ensure quality input in the predictive method. The hybrid method included an XGBoost algorithm. To tune the hyperparameters of the method, the WCS method was used adaptively. The integration of these hybrid systems improves the learning ability and forecast accuracy of the financial forecasting technique. The investigational outcomes showed that the WCS-XGBoost process outperformed most of the baseline methods, achieving accuracy with an RMSE and a MAPE. The implications of these findings support the available potential of utilizing evolutionary optimization-based hybrid systems in financial forecasting applications. Overall proposed flow as described in Figure 1.

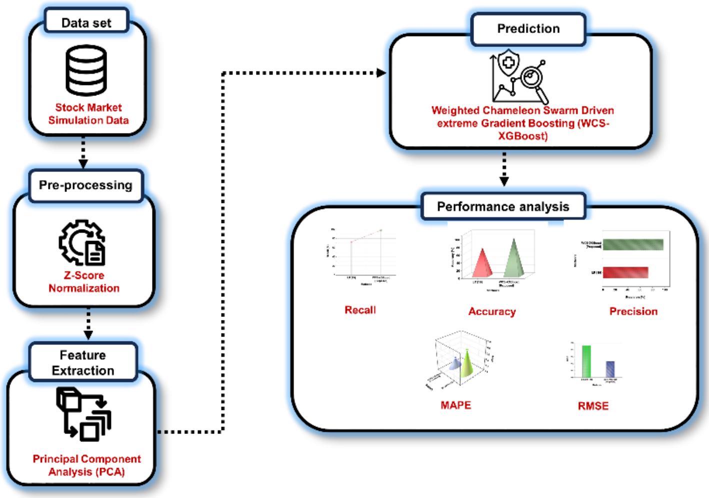

Figure 1: Workflow of the proposed framework for financial time series prediction

3.1 Dataset

The dataset for the stock market simulation is obtained from the Kaggle website [18]. It includes real stock market data generated with Geometric Brownian Motion to model value changes and Markov Chains for forecasting trends. The functionalities comprise forecasting time series data, modelling financials, and simulating algorithmic trading. The data characteristics are shown in figure 2.

Date	Company	Sector	Open	High	Low	Close	Volume	Market_C	PE_Ratio	Dividend_	Volatility	Sentimen	Trend
########	Uber	Technolog	100	101.0361	97.47781	100	171958	5.16E+11	24.25322	0.16309	0.047484	0.939232	Bearish
########	Tesla	Automoti	100.0711	102.038	97.15268	100.0711	196867	9.76E+11	18.60285	0.288515	0.022472	0.469417	Bearish
########	Panasonic	Finance	99.85783	101.5175	98.1081	99.85783	181932	4.6E+11	10.72893	2.221827	0.019991	0.399193	Bullish
########	Tencent	Automoti	98.85166	101.3038	96.99822	98.85166	153694	5.58E+11	14.5827	1.37774	0.036166	0.705853	Stable
########	Wells Farg	Automoti	98.39112	99.99049	96.23071	98.39112	169879	8.61E+11	37.49111	3.110198	0.03477	-0.76835	Stable
########	Snapchat	Consumer	99.31567	101.9511	97.61518	99.31567	160268	5.36E+11	28.65938	4.962211	0.023284	0.411912	Bearish
########	Adobe	Technolog	100.5501	102.752	98.64086	100.5501	104886	1.85E+11	22.36167	1.931616	0.047124	0.007377	Bullish
########	Oracle	Technolog	100.9013	103.5124	98.37995	100.9013	187337	3E+11	5.445021	2.869206	0.043794	0.658301	Stable
########	Novartis	Aerospace	101.0631	104.0734	98.83507	101.0631	137498	3.11E+11	39.73454	2.078019	0.018959	0.016283	Stable

Figure 2: Data features

3.2 Data preprocessing

After collecting the data, Z-score normalization is applied to continuous characteristics like stock prices, trading volume, and market capitalization. Z-score normalization, which is also referred to as standardization, transforms data to give each characteristic a mean of 0 and a Standard Deviation of 1 (Equation 1).

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Where x stands for the initial feature cost, μ represents the mean of the feature, σ represents the standard deviation of the feature, and z is the normalized value. Table 2 depicts the data preprocessing outcome and key metrics for model features.

Table 2: Outcome of pre-processed data

Date	Company	Sector	Open	High	Low	Close	Volume	Market Cap	PE Ratio	Dividend Yield	Volatility	Sentiment Score	Trend
2022-01-01	Uber	Technology	0.0000	0.2220	-0.1774	0.0000	1.3926	0.5537	0.6469	-0.7332	0.1491	1.1154	-1
2022-01-02	Tesla	Automotive	0.0001	0.3225	-0.6155	0.0001	1.7414	1.0419	-0.1787	0.7277	-0.9560	-0.8150	-1
2022-01-03	Panasonic	Finance	-0.0276	-0.0577	-0.0803	-0.0276	0.2516	-0.2975	-0.8866	1.4976	-1.2685	-0.4509	1
2022-01-04	Tencent	Automotive	-0.1327	-0.0895	-0.9753	-0.1327	-0.3941	0.0144	-0.6089	-0.0402	-0.3302	0.7675	0

Z-score normalization is important for the assurance that different features measured on different scales will contribute equally during model training. Z-score normalization speeds up convergence rates, creates a more stable model, and removes the emphasis on selected features with larger magnitudes (distance-based algorithms or gradient-based optimization).

3.3 Feature extraction using PCA

After pre-processing the data, PCA involves dimensionality reduction of the data by transforming the variables into uncorrelated components, preserving maximum variance to make it easier for analysis. PCA is active for feature extraction to decrease dimensionality and highlight the most important structures. It enhances data representation and system analysis by selecting relevant variables from large datasets in stock market data, enabling efficient analysis and extraction for predictive modelling. It reduces the total translation error; this transformation is carried out using PCA by identifying the $\emptyset$ feature of reflecting and vectorizing N-dimensional information into a package. Let's assume that v_S is a random dimensional N with a mean of ($\emptyset$), representing the input data recordings, Equation (2).

$$\emptyset = \frac{1}{N}\sum_{s=1}^{N} v_s \tag{2}$$

The definition of the covariance matrix is denoted in Equation (3), where N is the number of samples, E is determined by finding eigenvalues and eigenvectors of $(w_s - \mu)(w_s - \mu)^S$. Data factor levels highlight the most influential factors in content consumption.

$$E = \frac{1}{N}\sum_{s=1}^{N}(w_s - \mu)(w_s - \mu)^S \tag{3}$$

PCA solves the variance matrix DDu_s and an eigenvalue of λ_s using Equation (4), a threshold value that ensures the explained variance is sufficiently high. Contextual characteristics represent usage market patterns, including device and skill type.

$$DDu_s = \lambda_s u_s \tag{4}$$

The PCA technique identifies crucial features in eigenvalues and eigenvectors, resulting in a better representation of data. It determined how many principal components to select based on where the cumulative explained variance exceeds 95%. This allowed as much relevant information to be retained whilst reducing noise and confounding dimensions as much as possible.

3.4 Prediction using WCS-XGBoost

The WCS-XGBoost approach is applied to enhance time series prediction in financial market data. By applying WCS, the model accounts for dependencies in the data that are both short- and long-term, with XGBoost effectively handling non-linear relationships and interactions. This method improves forecasting accuracy and generalization, making it well-suited for predicting complex financial trends such as stock prices or market fluctuations.

3.4.1 eXtreme Gradient Boosting (XGBoost)

The XGBoost method is a gradient boosting method that combines decision trees and regularization to model nonlinear patterns for financial time series forecasting and to minimize errors in predictions. XGBoost is used to precisely forecast market trends, like stock prices or volatility, through efficient handling of large datasets and delivering strong, interpretable predictions. Gradient Boosting (GBoost) is less successful in noisy or high-dimensional data since it does not provide probabilistic outputs or uncertainty estimation. XGBoost is used to address difficulties in the GBoost technique with probabilistic regression. It simulates whole probability distributions and provides uncertainty estimates and more precise predictions, all essential for complex, real-time applications in the stock market. In general, XGBoost is a technique that seeks to estimate the parameters of a probability distribution$O_\theta\left(\frac{z}{w}\right)$. Oθzw, $w \in \mathbb{R}$ is the target variable, and $\theta \in \mathbb{R}^K$ is the distribution's parameter vector. The mean and the standard deviation involved assume a normal distribution. Three primary basics make up XGBoost: 1) a parametric probability distribution $e^{(m)}$, 2) the base learners $O_\theta\left(\frac{z}{w}\right)$, and 3) a suitable scoring rule $K(O_\theta, z)$. Shallow decision trees are used as foundation learners for the XGBoost algorithm in this architecture because of their practical efficacy and efficiency in real-time settings. The logarithmic score is expressed mathematically as Equation (5), which is the most often used scoring rule.

$$K(\theta, z) = -\log\left(O_\theta(z)\right) \tag{5}$$

Where $K(\theta, z)$ denotes the cost function, defined as the undesirable log-likelihood of observing a data point, $-\log\left(O_\theta(z)\right)$is the negative log-likelihood of z, in statistics and ML. XGBoost uses the natural gradient$\nabla^{\cdot} K\,(\theta, z)$ to learn the model parameters; it is driven by information geometry. More precisely, under the assumption that it has a training set $S = \{(w, z)\}_{j=1}^{N}$, it begins by estimating a common set of parameters ($\theta^{(0)}$) for the whole training set as follows in Equation (6).

$$\theta^{(0)} = argmin_\theta \sum_{j=1}^{N} K(\theta, z_j) \tag{6}$$

Where$\theta^{(0)}$represents the initial estimate, $argmin_\theta$ finds the value of θ that minimizes the following expression $\sum_{j=1}^{N} K(\theta, z_j)$, which denotes a summation over all N data points with relation to the predicted parameters up to that tree, Equation (7).

$$h_j^{(m)} = J_K(\theta_j^{(m-1)})^{-1}, \nabla_\theta K(\theta_j^{(m-1)}, z_j) \tag{7}$$

The optimization formula LF^* represents an optimal solution, and $O_{W,Z}(mg(W, Z) < 0)$ could be an objective function involving the parameters $W$and Z, which is represented in Equation (8).

$$LF^* = O_{W,Z}(mg(W, Z) < 0) \tag{8}$$

Where $J_K(\theta)$ denotes a generalized Jacobian approximation, depending on the context, $\nabla_\theta K(\theta, z) \nabla_\theta K(\theta, z)^S$ is the outer product of the gradient prediction. The following Equation (9) uses the preprocessed sensor input data to determine the necessary parameters for precise stock market data prediction.

$$o^{(m)} = argmin_o K(\theta_j^{(m-1)} - o.e^{(m)}(w_j), z_j) \tag{9}$$

Where $o^{(m)}$ represents the optimal step size at iteration m, $again_o$, this means to find the value of the o that minimizes the expression following it, $o.e^{(m)}(w_j)$ is a scaled step in the search direction, and the settings are modified using Equation (10).

$$\theta_j^{(m)} = \theta_j^{(m-1)} - \eta(o^{(m)} e^{(m)}(w_j)) \tag{10}$$

The XGBoost algorithm enhances financial market time series forecasting by detecting non-linear trends and offering probabilistic forecasts. It produces more precise, stable, and accurate forecasts, even under high-dimensional data conditions. Table 3 shows the hyper parameter settings for the XGBoost method in financial time series prediction.

Table 3: Hyperparameter settings for the XGBoost prediction method

Hyperparameter	Value
Learning Rate	0.1
Number of Estimators	100
Max Tree Depth	6
Subsample Ration	0.8
Column Sample by Tree	0.8

3.4.2 Chameleon swarm optimization (CSO)

Research will advance financial time series forecasting using Chameleon Swarm Optimization (CSO) to facilitate proper tuning of XGBoost. CSO emulates chameleons' adaptive hunting behavior to exploit the search space and optimally find the global search optimal parameter. The algorithm draws techniques inspired by the hunting and foraging behavior of chameleons, which are a special reptile capable of changing their color to camouflage in the surrounding environment. They are adaptable to many environments, including lowlands, deserts, and mountainous regions of the world. Chameleons mainly feed on insects. The hunting process consists of three phases: visual tracking, pursuit, and capturing prey (through targeted firing of their tongue). CSO suffers from premature convergence, poor exploitation-exploration balance, high computational overhead, dynamic computational cost, and the inability to adapt to complex high-dimensional optimization problems. WCS provides a better convergence behavior, stability, and adaptation to problems through adaptive weighting.

3.4.3 Weighted Chameleon Swarm (WCS)

WCS is applied to optimize XGBoost for accurate dynamic predictions in financial markets to utilize its bio-inspired swarm intelligence, mimicking a chameleon's adaptive behaviours. WCS optimizes by using adaptive weights to better explore the search space and the convergence, as well as to improve the exploration-exploitation balance and stability, in difficult shape prediction problems. Chameleons have super vision, which allows them to locate and follow prey in a variety of situations. It changes colours to match its environment, traverses through deserts, and climbs trees. The system simulates chameleons' natural hunting behaviour, including catching, chasing, and tracking.

Initialization and Function Evaluation: The initialization of WCS and its subsequent implementation are important components of modeling and optimizing predictive models to support stock market development. z_s^j defines its initial population in a search space with s dimensions using $z_{s,1}^j$, $z_{s,2}^j$ chameleons. Every chameleon shows a solution to the optimization problem of $k_i + q(v_i - k_j)$ has a position at a certain iteration z^j given by Equations (11) and (12).

$$z_s^j = [z_{s,1}^j, z_{s,2}^j, \dots z_{s,c}^j] \tag{11}$$

$$z^j = k_i + q(v_i - k_j) \tag{12}$$

Chameleon's Eyes Rotation: By rotating their eyes, chameleons determine where prey is located to optimize time series stock market data. The ability to turn aids in 360-degree prey detection; the subsequent actions take place as follows: The original position of the chameleon is the focus of severity (i.e., the start); the rotation matrix is used to detect the location of the prey; The chameleon's position is updated using a rotation matrix at the gravitational midpoint, and then returned to its initial position to refine market stage optimization. $z_{s+1}^{j,i}$ updates the position at the $s + 1$ iteration and $z_s^{j,i} + O_1(O_s^{j,i} - H_s^i)$ is global solution. $z_s^{j,i} + \mu(v^i - k^i)q_3 + k_a^i$ denotes the positive control parameters, which are presented in Equation (13).

$$z_{s+1}^{j,i} = \begin{cases} z_s^{j,i} + O_1(O_s^{j,i} - H_s^i)q_2 + O_2(H_s^i - z_s^{j,i})q_1 & q_1 \geq O_o \\ z_s^{j,i} + \mu(v^i - k^i)q_3 + k_a^i dgn(rand - 0.5)q_1 & q_1 < O_o \end{cases} \tag{13}$$

Hunting of prey: The WCS algorithm's prey behaviour hunting is essential for reducing the distribution of resource deployment for financial market building construction. It optimizes the model parameters, efficiently handling complex market data. More informed and effective decision-making in financial strategies can be enabled using Equation (14).

$$u_{s+1}^{j,i} = xu_s^{j,i} + d_1(H_s^i - z_s^{j,i}) + d_2(O_s^{j,i} - z_s^{j,i})q_2 \tag{14}$$

Where, $u_{s+1}^{j,i}$ denotes the updated position of the agent at step $s + 1$in the swarm that represents the agent's new state after movement. d_1 and d_2 are weight factors that control the influence of the prey hunting behaviour. WCS algorithm modifies the location of agents within the swarm, optimizing their waves towards more favorable locations (prey or optimum points), allowing for better resource allocation in financial market strategies. Table 4 illustrates the hyperparameter scenarios for the WCS optimization method. Pseudocode 1 shows the proposed WCS-XGBoost method's working procedure.

Table 4: Hyperparameter scenarios for the optimization method

Hyperparameter	Value
Population Size	30
Max Iterations	100
Prey Behavior Weights (d1, d2)	0.1 – 0.9
Step Size Learning Rate (η)	0.05

Pseudocode 1: Weighted Chameleon Swarm-driven eXtreme Gradient Boosting (WCS-XGBoost)

$Start$

$Step$ 1: $Load\ and\ preprocess\ time\ series\ data$

$Input$: $Financial\ time\ series\ dataset\ D = \{(w\_j, z\_j)\}\ for\ j = 1\ to\ N$

$Apply\ Z - score\ normalization$

$Apply\ PCA\ for\ dimensionality\ reduction$

$Step$ 3: $Train\ the\ final\ XGBoost\ model\ using\ optimized\ \theta\_best$

$For\ each\ data\ point\ j$:

$Calculate\ the\ scoring\ function\ using$ Equation (5)

$Step$ 5: $After\ optimization, extract\ the\ best\ parameters\ \theta\ best$ using Equation (6)

$Update\ the\ gradient\ direction\ using$ Equation (7)

$Solve\ optimization\ using$ Equation (8)

$Compute\ the\ optimal\ step\ size\ using$ Equation (9)

$Update\ parameter\ \theta\ using$ Equation (10)

$Step$ 4: $Begin\ optimization\ loop$

$For\ iteration\ s = 1 to\ max\_iter$:

$For\ each\ agent\ j$:

$Evaluate\ current\ fitness$: $k_j = -MSE(XGBoost\_model(z^j))$

$Eye\ rotation\ mechanism$

$Step$ 2: $Initialize\ WCS\ parameters$

$Initialize\ swarm\ population\ z_s^j \in \mathbb{R}s\ for\ each\ chameleon\ agent\ j\ using$ Equation (11)

$For\ each\ agent\ j\ in\ the\ swarm$:

$Initialize\ position\ using$ Equation (12)

$Initialize\ velocity\ v_i\ randomly$

$Evaluate\ fitness\ k_j = -MSE(XGBoost\_model(z^j))$

$If\ rand < q_1$:

$Update\ position\ using$ Equation (13)

$Else$:

$Prey\ hunting\ behavior$

$Update\ velocity\ using$ Equation (14)

$Update\ position$

$z^j = z^j + z_{s+1}^{j,i}$

$Evaluate\ new\ fitness$

$new\_fitness = -MSE(XGBoost\_model(z^j))$

$If\ new\ fitness > f\_j$:

$Accept\ the\ new\ position$

$Else$:

$Revert\ to\ the\ previous\ best\ position$

$End\ function$

WCS-XGBoost utilizes a swarm intelligence optimization method to advance the accuracy of predictions for complex time series scenarios, making the model more suitable, robust, and well-performing in difficult financial forecasting tasks. WCS-XGBoost differs from other hybrid models combining boosting and metaheuristic--it types bio-inspired adaptation and bio-inspired adaptation with PCA dimensionality reduction--to develop an optimized, full hierarchical time series forecasting pipeline designed specifically for nonlinear financial data. The WCS-XGBoost method, which combines the XGBoost method with the WCS optimization strategy, is a powerful combination. This combination greatly improves the methods hyperparameter optimization process which allows it to escape local optimum and achieve better generalization. WCS's adaptive and exploratory exploitation strategy allows for easier, faster, and more stable convergence during training. As a result of WCS-XGBoost's great predictive accuracy by being generally better than methods that are conventional and other hybrid methods, this is especially true for highly complex nonlinear financial time series data. Furthermore, WCS-XGBoost is robust, consistent, and capable of handling various data distributions allowing practical forecasting situations to be undertaken by the model.

4 Result and discussion

The aim is to develop an effective time series prediction model for financial market data to improve stock data forecasting accuracy and support strategic decision-making under uncertainty. Table 5 shows the experimental setup, and Table 6 displays the ablation study on the impact of each component in the WCS-XGBoost framework for financial time series prediction.

Table 5: Experimental setup

Components	**Specification**
Operating System	Windows 10
Graphics Card (GPU)	NVIDIA GeForce RTX 3060
Processor (CPU)	Intel Core i7
Software	Python 3.x, pandas, NLTK, TensorFlow/Keras, Flask
Storage	500 GB SSD
RAM	16 GB DDR4

Table 6: Ablation study on the impact of each component in the WCS-XGBoost framework for financial time series prediction

Method Variant	Accuracy (%)	Precision (%)	Recall (%)
Data + Z-Score Normalization + PCA	88	87	86
Data + Z-Score Normalization + PCA + XGBoost	90	89	88
Data + Z-Score Normalization + PCA + XGBoost + CSO	93	92	91
Data + Z-Score Normalization + PCA + XGBoost + WCS	95	94	93
Data + Z-Score Normalization + PCA + XGBoost + WCS + Feature Tuning	**98.69**	**98**	**98**

ROC: The ROC curve graphically illustrates how well a classification model performs; it evaluates the discriminating model, and the TPR and FPR are compared across various thresholds. Figure 3 displays the outcome of the ROC curve.

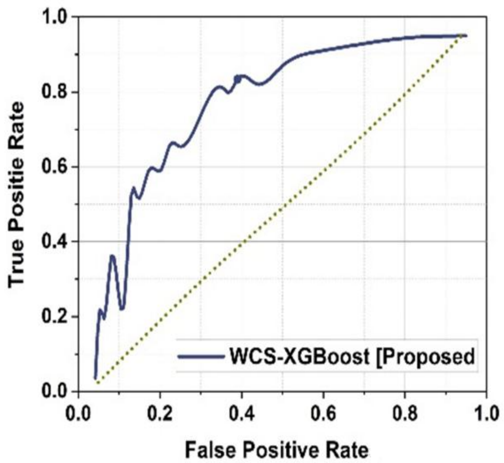

Figure 3: Graphical representation of ROC

The steady increase in method accuracy with more training epochs indicates that WCS-XGBoost effectively learns from the data and progressively improves its performance.

4.1 Stock market prediction

WCS-XGBoost predicts stocks into predefined categories using learned representations from historical data. It incorporates technical indicators, such as sentiment analysis, and fundamental metrics to enhance performance on financial time series. Figure 4 shows the Stock Market Prediction. This model's close tracking of the actual stock price indicates significant forecasts for stock trends, although slight deviations show the very nature of stock price predictions.

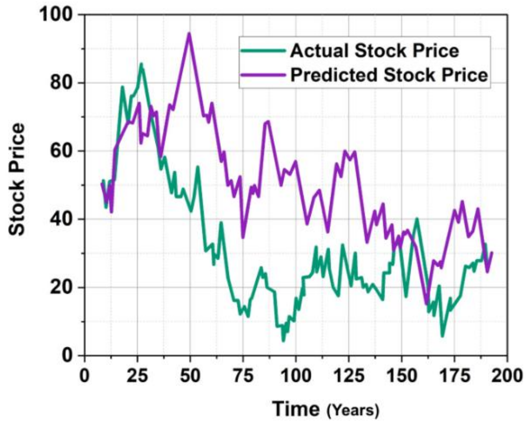

Figure 4: Stock market prediction

Comparative analysis : The evaluation of the effectiveness of the WCS-XGBoost method is associated with existing methods, including BiLSTM [20] and LR [19]. The comparative analysis uses multiple performance metrics, recall, accuracy, precision, RMSE, and MAPE to provide a comprehensive assessment of model efficacy. In this research, parameter choices for the WCS-XGBoost method were based on maximizing predictive performance across critical classification performance metrics (accuracy, precision, and recall) that are inherent to reliable stock market forecasting. The result is summarized in Table 7.

Accuracy is the degree to which a measurement or computation closely matches the real value of observations and corresponds to an accurate value.

Precision is the degree of precision in calculating the potential between measurements of the same element. **Recall** reflects a method's ability to correctly identify all relevant instances within a given dataset. Figure 5 illustrates the result of accuracy.

Table 7: Numerical outcome of metrics

Methods	Accuracy (%)	Precision (%)	Recall (%)
LR [19]	72.7	72.9	72.7
WCS-XGBoost [Proposed]	**98.69**	**98.02**	**98.00**

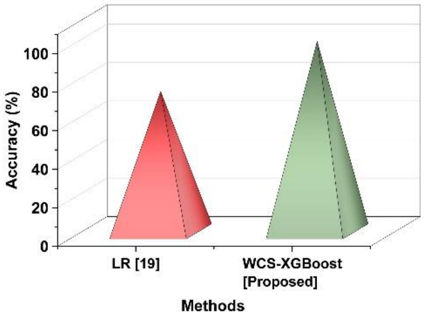

Figure 5: Result of accuracy

The proposed WCS-XGBoost method achieved an accuracy of 98.69%, significantly outperforming the existing LR approach, which attained only 72.7%. This substantial improvement underscores the efficiency and reliability of the proposed strategy. Figure 6 presents the comparative precision outcomes, further highlighting the method's predictive advantage.

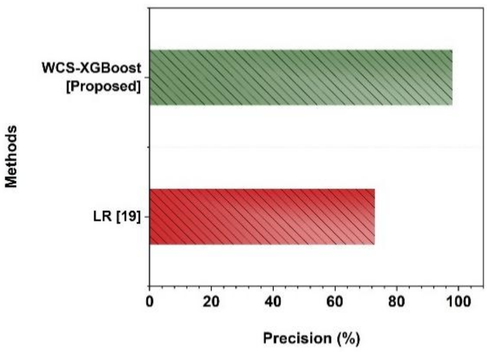

Figure 6: Precision comparison between the proposed method and the LR method [19]

Compared to the existing method, WCS-XGBoost achieved a precision 0f 98.02%, notably surpassing LR, which is 72.9%. This indicates that the proposed methods enhanced effectiveness in producing accurate predictions. Figure 7 illustrates the recall performance, further confirming its superiority.

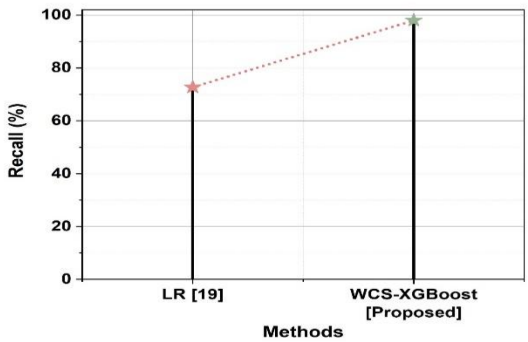

Figure 7: Performance outcome of recall

WCS-XGBoost achieved a significantly higher recall of 98.00% compared to 72.7% with LR, highlighting its superior ability to accurately identify relevant financial patterns and outperform conventional models.

RMSE calculates the mean magnitude of prediction errors in a model, applying a severe penalty for greater errors due to the squaring of the modification between predicted outcomes and true observations. **MAPE** refers to the different percentage deviation between forecasted results and observed values, offering relative indicators of forecast accuracy. Table 8 shows the numerical results of error metrics. Figure 8 shows the evaluation of (a) RMSE and (b) MAPE.

Table 8: Numerical results of error metrics

Methods	RMSE	MAPE
BiLSTM [20]	0.4605	0.795
WCS-XGBoost [Proposed]	**0.2312**	**0.321**

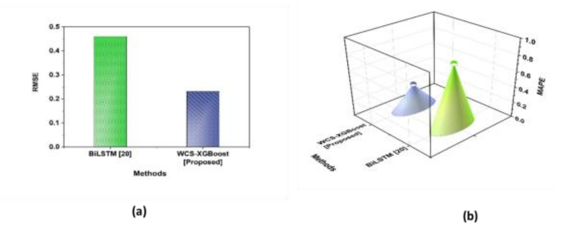

Figure 8: Evaluation of (a) RMSE and (b) MAPE

The BiLSTM performed moderately with an RMSE of 0.4605 and an MAPE of 0.795. The WCS-XGBoost model outperformed both, with much lower RMSE (0.2312) and MAPE (0.321). These findings demonstrate that WCS-XGBoost outperforms existing approaches as a robust and adaptable framework for analyzing market trends and guiding financial decision-making.

4.2 Dataset comparison

In this research, the proposed WCS-XGBoost model is trained in both financial dataset such as Historical Stocks Price Dataset [21] and Stock Market Simulation Dataset. The goal of this comparative evaluation was to investigate the generalization capability of the proposed model in both synthetic and real market environments. The method's presentation was estimated using standard classification evaluation metrics: recall, accuracy, and precision. As shown in Table 9, the WCS-XGBoost model performed consistently well against both datasets, albeit with slightly better performance on the simulation dataset. This small, though meaningful difference, was a strong indication that the model is robust and sufficiently adaptable to learn complex financial patterns, irrespective of the complexity of the underlying data generation process. All of these results help validate the model's utility prospectively for real-world application, in financial forecasting, where a small performance improvement can have a considerable economic impact.

Table 9: Performance comparison of the proposed WCS-XGBoost model in simulation and real-world stock market datasets

Metrices	Stock Market Simulation Dataset [Proposed]	Stocks Historical Price Dataset [21]
Accuracy	98.69	98.62
Precision	98.02	97.58
Recall	98.00	97.93

The proposed WCS-XGBoost model has shown a better predictive performance when compared with the Stocks Historical Price Dataset [21], especially with the metrics used for evaluation. For example, on the stock market simulation dataset evaluated, accuracy was 98.69%, which is greater than the accuracy of the historical stock price model (98.62%). Likewise, precision (98.02%) and recall (98.00%) were compared with 97.58% and 97.93% respectively, from the model in [21]. Although the differences in accuracy, precision, and recall were only small in percentage, they are important for financial forecasting based on stock market predictions, since minor improvements can lead to more accurate decision-making. These results indicate the proposed method generalizes well between different datasets and consistently performs well in estimating stock market behavior. The use of PCA for dimensionality reduction and CSO for dynamic hyperparameter tuning enhanced the robustness and accuracy of the proposed framework when associated with the previous method [21] for treating and interpreting financial time series data with its nonlinear complexities.

4.3 Cross validation

Cross-validation is a type of model evaluation process that separates parts of the data into different subsets to examine the model's performance stability, generalization, and overfitting avoidance. This research conducted a 5-fold cross-validation and applied cross-validation to the results of WCS-XGBoost, ensuring that WCS-XGBoost produced similar consistent performance and was a reliable model across subsets of the training and unseen stock market dataset. Table 10 illustrates the 5-fold cross-validation results for WCS-XGBoost.

Table 10: Performance metrics of the WCS-XGBoost model using 5-fold cross-validation in financial time series data

Fold	Accuracy (%)	Precision (%)	Recall (%)	RMSE	MAPE
1	98.55	97.90	97.80	0.235	0.325
2	98.70	98.10	98.00	0.232	0.320
3	98.60	97.85	97.90	0.234	0.324
4	98.75	98.20	98.10	0.229	0.318
5	98.85	98.05	98.10	0.226	0.316
Average	**98.69**	**98.02**	**98.00**	**0.2312**	**0.321**

The performance robustness and generalization ability of the WCS-XGBoost model introduced in this research were evaluated using 5-fold cross-validation on the stock market simulation dataset. For each fold, the dataset was separated into training and validation subsets to mitigate overfitting and make fair estimates of performance. The dataset was separated into 5-folds, and evaluation metrics such as recall, accuracy, precision, RMSE, and MAPE were calculated. The WCS-XGBoost method achieved high performance across all folds. All metrics are presented in Table 10. WCS-XGBoost had an average accuracy of 98.69%, precision of 98.02%, and recall of 98.00% across all folds. All error metrics were very low for the lifespan of the prediction across folds, with RMSE of 0.2312 and MAPE of 0.321 providing a consistency throughout different splits of the data, showing the WCS-XGBoost model is reliable and stable.

The research developed an effective time series prediction model for financial market data to enhance stock price forecasting. There were four problems with the study: it did not have integration with real-time data sources, there was limited applicability by context through industry, there was reduced explainability, and applied validation efforts across enterprise scales and contexts were minimal [8]. The model overfitted on very volatile datasets, had limited flexibility to react to changing market conditions, and lacked generalizability over a broad slice of financial regimes [9]. Clustering methods struggled with financial data dimensionality, lacked precision for temporal dependencies, and produced less interpretable clusters [10]. Had problems validating across different financial sectors, real-time dexterity, interpretability in black-box models, and large institutional scale (Bu). There was a lack of generalizability to non-chain markets, reduced interpretability, a lack of sufficient multi-source data integration, and increased computational requirements for dynamic updates (Jio). The WCS-XGBoost system was compared with conventional systems such as LR [19] and BiLSTM [20]. LR [19] has complexity in managing high-dimensional data and was susceptible to kernel regression assortment in the stock market. The systems were prone to over-fitting and needed considerable training data, limiting their relevance in complex market analysis methods. BiLSTM [20] overfits because of its complexity, particularly when trained on restricted, highly detailed real-world market data. In contrast, the WCS-XGBoost technique efficiently overcomes these restraints by capturing nonlinear connections, directing feature significance, and making more robust and precise stock market forecasts.

5 Conclusion

Financial time series prediction is vital to facilitate thoughtful investment decision-making, risk management, and economic forecasting. This research outlined a new hybrid model that synthesizes WCS with XGBoost to provide improved forecasting for complex and nonlinear financial data. The dataset consists of a variety of financial variables, including foreign exchange rates, stock indices, and prices on commodities. Data preprocessing involved Z-score normalization as a standard input feature normalization technique and PCA as a data dimensionality reduction technique to clean up and improve our input data. The use of WCS provided intelligent hyperparameter tuning for XGBoost through adaptive weights on the TD learning; in addition, an intelligent exploration-exploitation balance will lead to faster convergence. The WCS-XGBoost model outperformed the baseline model by a substantial margin, achieving an accuracy of 98.69%, a recall of 98.00%, a precision of 98.02%, MAPE (0.321, and RMSE (0.2312, suggesting that the model is robust across the financial domains. The ablation study also demonstrated the utility of all components, namely PCA and WCS, which work to improve model performance. The research did not consider real-time data handling, estimation of uncertainty, and integration of multiple data sources, and this restricts adjustment in highly volatile financial circumstances. In future research, it will incorporate real-time data streaming, uncertainty estimation, and data fusion from multiple different sources to further enhance the robustness of the prediction. The proposed system is expected to support a reliable, scalable, and data-driven approach to financial forecasting and offer some practical value to financial analysts, traders, and policymakers working in fast-moving financial markets. The WCS-XGBoost method will be validated with real-world financial datasets obtained for its robustness and generalizability beyond those that were achieved in simulated datasets for the purposes of this research.

References

[1] Zheng H, Wu J, Song R, Guo L, &Xu Z (2024). Predicting financial enterprise stocks and economic data trends using machine learning time series analysis. *Applied and Computational Engineering*, 87, 26-32. https://doi.org/10.54254/2755-2721/87/20241562

[2] Sharma R, & Mehta K (2024). Stock market predictions using deep learning: developments and future research directions. *Deep Learning Tools for Predicting Stock Market Movements*, 89-121. https://doi.org/10.1002/9781394214334.ch4

[3] Gong H (2024). An Enhanced Hybrid Model for financial market and economic analysis: a case study of the Nasdaq Index. *International Journal of System Assurance Engineering and Management*, 15(7), 3406-3423. https://doi.org/10.1007/s13198-024-02349-0

[4] Zareeihemat P, Mohamadi S, Valipour J, &Moravvej S.V (2025). Forecasting stock market volatility using housing market indicators: A reinforcement learning-based feature selection approach. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3554224

[5] Yang C.H, Molefyane T, Lee B, Hsueh T, & Lin Y.D (2025). Improving Investment Forecasting: A Comparative Analysis of Machine Learning Models as Key GDP Indicators. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3554641

[6] Amiri B, Haddadi A, &Mojdehi K.F. (2025). A Novel Hybrid GCN-LSTM Algorithm for Energy Stock Price Prediction: Leveraging Temporal Dynamics and Inter-Stock Relationships. *IEEE Access*. https://doi.org/10.1109/ACCESS.2025.3536889

[7] Pan H, Tang Y, & Wang G (2024). A Stock Index Futures Price Prediction Approach Based on the MULTI-GARCH-LSTM Mixed Model.

Mathematics, 12(11), 1677. https://doi.org/10.3390/math12111677

[8] Zhang S (2025). A Big Data-Driven Approach to Financial Analysis and Decision Support System Design. Informatica, 49(11). https://doi.org/10.31449/inf.v49i11.7065

[9] Bhambu A, Gao R, &Suganthan P.N. (2024). Recurrent ensemble random vector functional link neural network for financial time series forecasting. *Applied Soft Computing*, 161, 111759. https://doi.org/10.1016/j.asoc.2024.111759

[10] Cortés D.G, Onieva E, López I.P, Trinchera L, & Wu J (2024). Autoencoder-Enhanced Clustering: A Dimensionality Reduction Approach to Financial Time Series. *IEEE Access*, 12, 16999-17009. https://doi.org/10.1109/ACCESS.2024.3359413

[11] Durairaj D.M., & Mohan B.K. (2022). A convolutional neural network-based approach to financial time series prediction. *Neural Computing and Applications*, *34*(16), pp.13319-13337. https://doi.org/10.1007/s00521-022-07143-2

[12] Li Y, Chen L, Sun C, Liu G, Chen C, & Zhang Y (2024). Accurate stock price forecasting based on deep learning and hierarchical frequency decomposition. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3384430

[13] Lazcano A, Herrera P.J, &Monge M (2023). A combined model based on recurrent neural networks and graph convolutional networks for financial time series forecasting. *Mathematics,* 11(1), 224. https://doi.org/10.3390/math11010224

[14] Albahli S, Irtaza A, Nazir T, Mehmood A, Alkhalifah A, &Albattah W (2022). A machine learning method for the prediction of the stock market using real-time Twitter data. *Electronics*, 11(20), 3414. https://doi.org/10.3390/electronics11203414

[15] Bu Y (2024). Fuzzy Decision Support System for Financial Planning and Management. Informatica, 48(21). https://doi.org/10.31449/inf.v48i21.6718

[16] Jiao Z (2024). Dynamic Financial Distress Prediction Using Combined LASSO and GBDT Algorithms. Informatica, 48(17), pp.139-152. https://doi.org/10.31449/inf.v48il7.6493

[17] González-Núñez E, Trejo LA, and Kampouridis M (2024). A comparative study of stock market forecasts based on a new machine learning model. Big Data and Cognitive Computing, 8(4), p.34. https://doi.org/10.3390/bdcc8040034

[18] Dataset: https://www.kaggle.com/datasets/samayashar/stock-market-simulation-dataset

[19] Mokhtari S, Yen K.K, & Liu J (2021). Effectiveness of artificial intelligence in stock market prediction based on machine learning. arXiv preprint arXiv:2107.01031. https://doi.org/10.5120/ijca2021921347

[20] Yang M, & Wang J (2022). Adaptability of financial time series prediction based on BiLSTM. *Procedia Computer Science*, 199, 18-25. https://doi.org/10.1016/j.procs.2022.01.003

[21] Dataset: https://www.kaggle.com/datasets/prodzar/stocks-historical-price-data

Acronyms	**Description**
TN	True Negative
TP	True Positive
FP	False Positive
FN	False Negative
LSTM	Long-term and short-term memory
RedRVFL	Recurrent Ensemble Deep Random Vector Functional Link
DL	Deep learning
CNN	Convolutional Neural Networks
GCN	Graph Convolutional Networks
BiLSTM	Bidirectional Long Short-Term Memory
ELM	Extreme Learning Machine
LLMs	large language models
TPR	true positive rate
FPR	false positive rate
ROC	Receiver Operating Characteristic